8.0 - 1 0.2 6.5
JSC-14457
NASA CR-160717

DESIGN SPECIFICATION FOR A LIST PROCESSING SYSTEM

JOB ORDER 71-593

(TIRF 78-0021)

(E80-10265) DESIGN SPECIFICATION FOR A LIST PROCESSING SYSTEM (Lockheed Electronics Co.) 77 p HC A05/MF A01 CSCL 05B

N80-29812

Unclas
G3/43 00265

Prepared By

Lockheed Electronics Company, Inc.

Systems and Services Division

Houston, Texas

Contract NAS 9-15200

For

EARTH OBSERVATIONS DIVISION

SCIENCE AND APPLICATIONS DIRECTORATE

National Aeronautics and Space Administration

LYNDON B. JOHNSON SPACE CENTER

Houston, Texas

August 1978

LEC- 12696

JSC-14457

DESIGN SPECIFICATION
FOR A
LIST PROCESSING SYSTEM

(TIRF 78-0021)

JOB ORDER 71-593

Prepared by

C. L. Horton
C. W. Ahlers
P. J. Aucoin, Jr.

APPROVED BY

John A. Rainey

John A. Rainey, Supervisor
Scientific Applications Section

J. C. Minter for

B. L. Carroll, Manager
LACIE Development and Evaluation Department

Prepared By
Lockheed Electronics Company, Inc.
For
Earth Observations Division

August 1978 LEC-12696

CONTENTS

Section		Page
1.	SCOPE	1-1
2.	APPLICABLE DOCUMENTS	2-1
3.	SYSTEM DESCRIPTION	3-1
	3.1 HARDWARE	3-1
	3.2 SOFTWARE DESCRIPTION	3-1
	3.2.1 SOFTWARE COMPONENT NO. 1 (LISTPR EXEC)	3-2
	3.2.2 SOFTWARE COMPONENT NO. 2 (DAMRG)	3-10
	3.2.3 SOFTWARE COMPONENT NO. 3 (GTTCN)	3-11
	3.2.4 SOFTWARE COMPONENT NO. 4 (GTDDM)	3-12
	3.2.5 SOFTWARE COMPONENT NO. 5 (PREPPT)	3-13
	3.2.6 SOFTWARE COMPONENT NO. 6 (MPPTA)	3-15
	3.2.7 SOFTWARE COMPONENT NO. 7 (MPPTC)	3-17
	3.2.8 EOD-LARSYS SYSTEM - DISPLAY PROCESSOR	3-18
4.	OPERATION	4-1
Appendices		
A.	SUBPROGRAM DESCRIPTIONS FOR PREPPT	A-1
B.	SUBPROGRAM DESCRIPTIONS FOR MPPTA	B-1
C.	SUBPROGRAM DESCRIPTIONS FOR MPPTC	C-1
D.	LIST VARIABLE INPUT EQUATIONS	D-1

1. SCOPE

This document contains a preliminary design specification for a LIST (LABEL IDENTIFICATION FROM STATISTICAL TABULATION) processing system. This system provides for dot labeling using quadratic discrimination based on analyst-provided answers to questions concerning an MSS pixel array (segment) and individual dots (pixels) in the segment.

The system consists of several subsystems which communicate through disk and tape files. These subsystems are

1. The file merge processor of the EOD-LARSYS system, (DAMRG)
2. The ground truth to class map and class map to labeled dots processors of the EOD-LARSYS system, (GTTCN, GTDDM)
3. The LIST variable processor, (PREPPT)
4. The discriminator training program, (MPTTA)
5. The discriminator classification program, (MPTTC)
6. The EOD-LARSYS Procedure 1 system modified to accept Type dots re-labeled as boundary dots and output CCIT-type comparison reports.

The LIST processing system will be written in FORTRAN IV-G and implemented on the IBM 370/148 computer at Purdue-LARS.

2. APPLICABLE DOCUMENTS

- As-Built Design Specification for a merging Program for formatted Image Data Files, Lockheed Electronics Company, Inc., SSD, Houston, Texas, Aug 1978, LEC-12653.
- Preliminary User Guide for the Program GTDDM (Ground Truth Dot Dumps, Lockheed Electronics Company, Inc., SSD, Houston, Texas, July 1978, LEC-12636.
- "As-Built" Design Specification for the Patterson-Pitt-Thandani Minimum Loss Classifier, Lockheed Electronics Company, Inc., SSD, Houston, Texas, May 1978, LEC-12285.
- Preliminary User Guide for the Program GTTCN (Ground Truth Tape Conversion) LEC, H,T, July 1978, LEC-12635.
- "As-Built" Design Specification for Lacie Formatted Dot Cards in EOD-LARSYS, LEC, April 1978, LEC-12154.
- As-Built Specification for EOD-LARSYS Procedure 1, TIRF 77-0008, LEC-11293, JSC-13143, Oct. 1977.

3. SYSTEM DESCRIPTION

3.1 HARDWARE

The LIST processing program is operational on the IBM 370-148 at Purdue LARS under the CMS370 operating system. The program utilizes the IBM Fortran IV-G compiler.

3.2 SOFTWARE DESCRIPTION

The LIST processing system features a semi-automatic dot labeling procedure. The quadratic discriminator is trained based on analyst-furnished responses to questions concerning the segment under investigation, and individual dots, raw MSS data and ground truth dot labels. This discriminator is then used in the test and classification phase to label dots utilizing the analyst-supplied responses.

The complete system consists of two main parts. The first part is the LIST labeling procedure. The second part utilizes the LIST labeled dots in conjunction with the EOD-LARSYS Procedure 1 system for classification of segment pixels.

3.2.1 SOFTWARE COMPONENT NO. 1 (LISTPR EXEC)

The LIST executive file requests the tapes for each procedure and releases them when they are no longer needed, loads and starts execution of each processor, defines input and ouput files, saves intermediate files and prints list files.

3.2.1.1 Linkages

The LIST executive uses the FORTRAN-IVG compiler, routine CMS370 software systems routines, data files and the following processor programs: DAMRG, GTTCN, GTDDM, PREPPT, MPPTA, MPPTC and the EOD-LARSYS system.

3.2.1.2 Interfaces

Interface is accomplished through the following files:

16-channel Data File in EOD-LARSYS format.
Ground Truth Class Map in Universal format.
LIST Variable Input File (Table 3-5).
Loss Vector Matrix File.

3.2.1.3 Inputs

Inputs to the LIST system are described for each of the processors DAMRG, GTTCN, GTDDM, PREPPT, PPTA and PPTC. The master LIST control file is available to each processor for the types of processing (TRAIN, TEST, CLASSIFY), segment numbers and dates.

3.2.1.4 Outputs

Outputs from the LIST processor are described for each of the subprocessors DAMRG, GTTCN, GTDDM, PREPPT, PPTA and PPTC.

3.2.1.5 Storage Requirements

The LIST processor requires less than 1024k bytes of virtual memory.

3.2.1.6 Description

The LIST processor reads input control data giving the segment numbers to be processed and the type of processing to be done. If requested the subprocessor Merging Formatted Image Data Files, DAMRG, is called to produce a 16-Channel Tape from portions of 4 four-channel tapes. This tape consists of one logical file for each segment.

If training or testing is to be done, the Ground Truth Tape Conversion subprocessor, GTTCN, is called to produce the Ground Truth Class Map Disc File from the one-channel Ground Truth Tape.

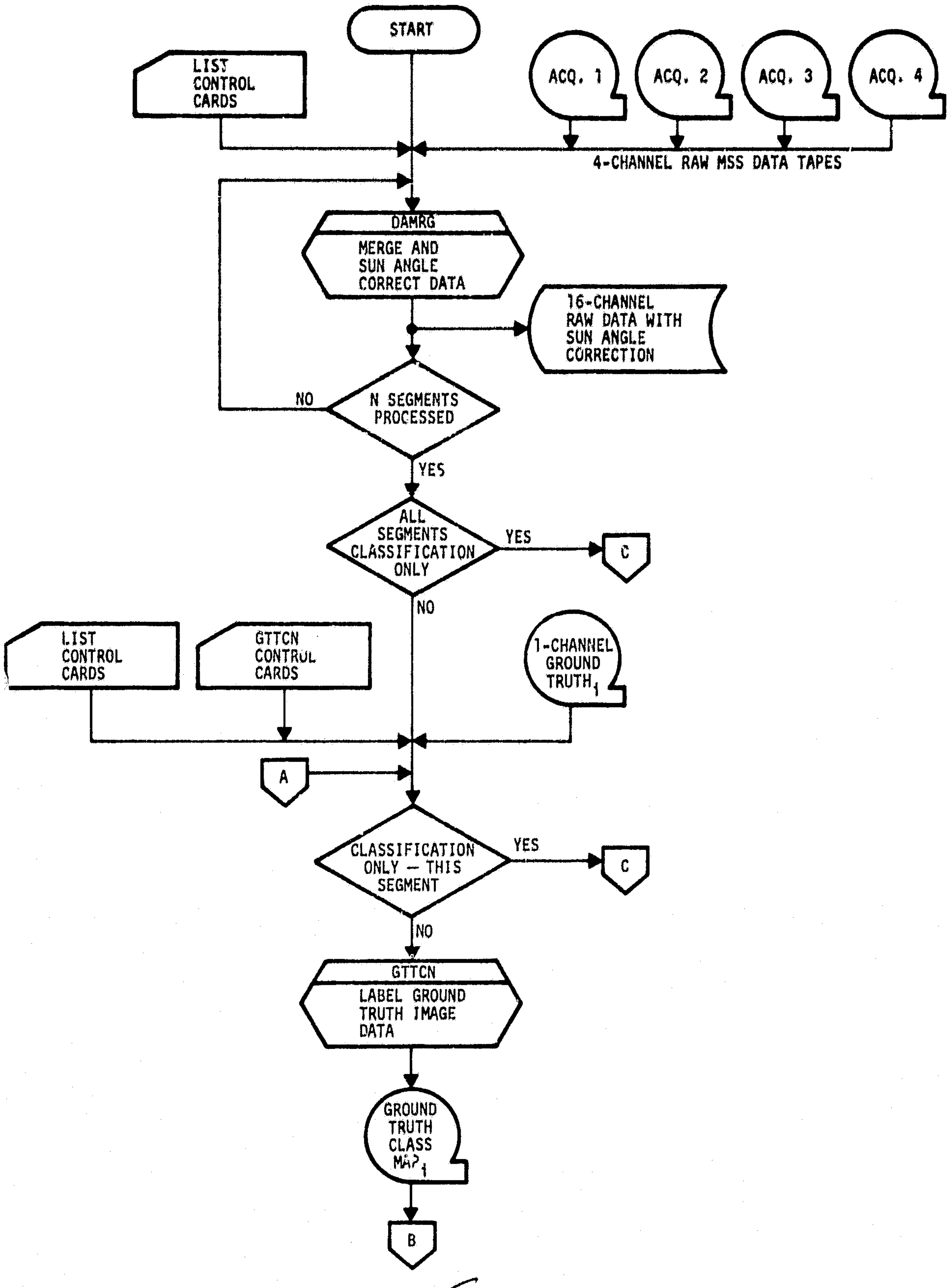
This disc file is then read by the Ground Truth Dot Dump subprocessor, GTDDM to produce the Ground Truth Dot File also consisting of one logical file for each segment.

If requested, the Preprocessor for the Patterson-Pitt-Thandani Routines, PREPRT, is called to prepare data for the quadratic discriminator. The 16-Channel Data File created by DAMRG, the AI Responses File, and, if training is to be done, the Ground Truth Dot Data File are used with the Seasonal Table and the Biostage Response Table to produce the LIST Variable Input File.

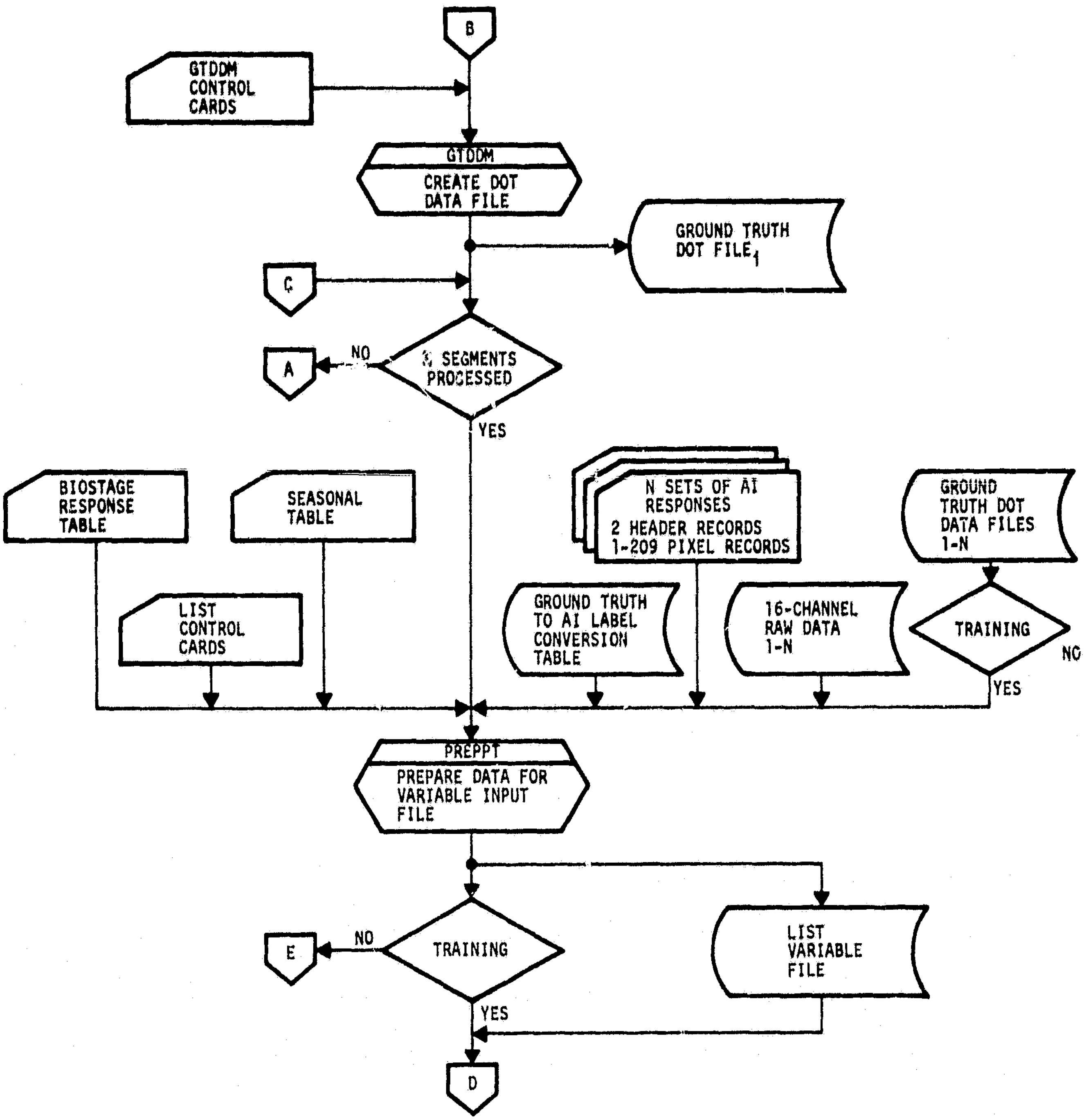
If requested, the Patterson-Pitt-Thandani Training Processor, MPPTA, is called to create the loss vector matrix, and the Patterson-Pitt-Thandani Classifying Processor, MPPTC, is called to calculate the classification losses to find the minimum loss and write the computed dot labels (classification results).

If requested, the EOD-LARSYS System Display Processor is called the classify the entire scene using the 16-Channel Raw Data File created by DAMRG, the EOD-LARSYS control cards, the Analyst Changes to Computed Dot Labels File and if type is TEST the Ground Truth Dot Labels File. This classification produces the Class Map File, the Dot Summaries and Proportion Estimate File and a Dot Summary Listing. If type is TEST, this processor will also produce a file comparing the classification results to ground truth.

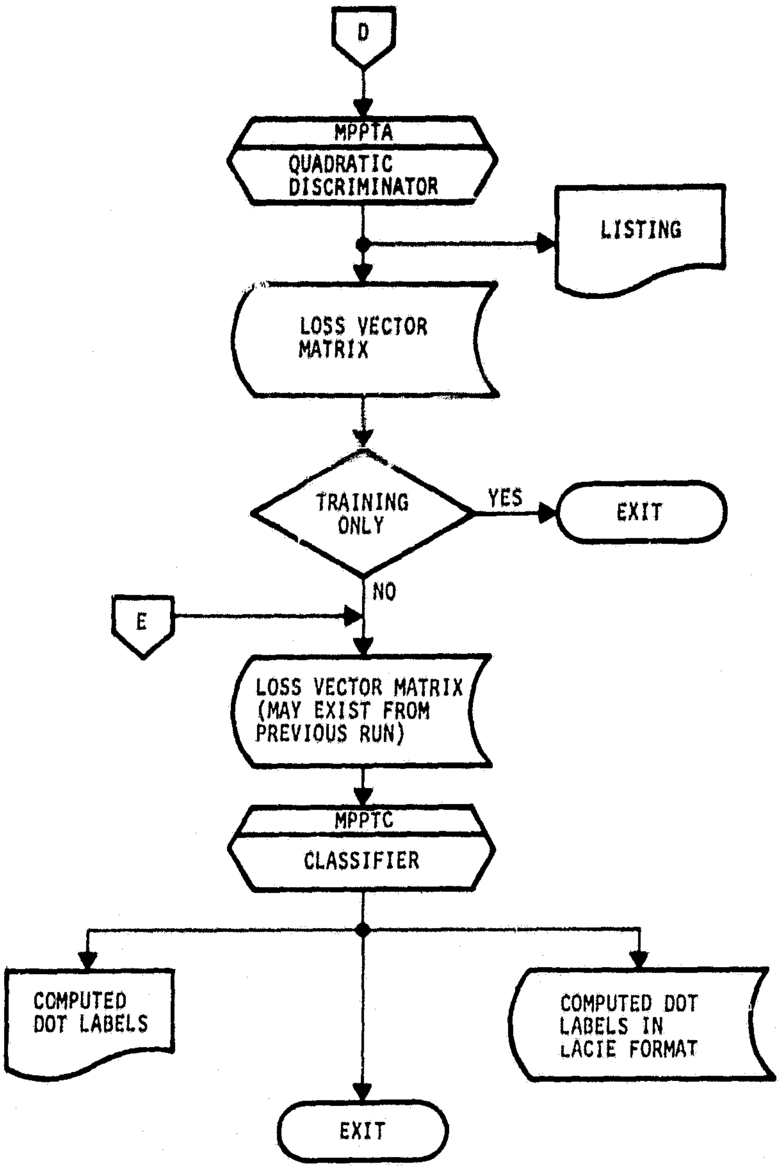
LIST PROCESSOR



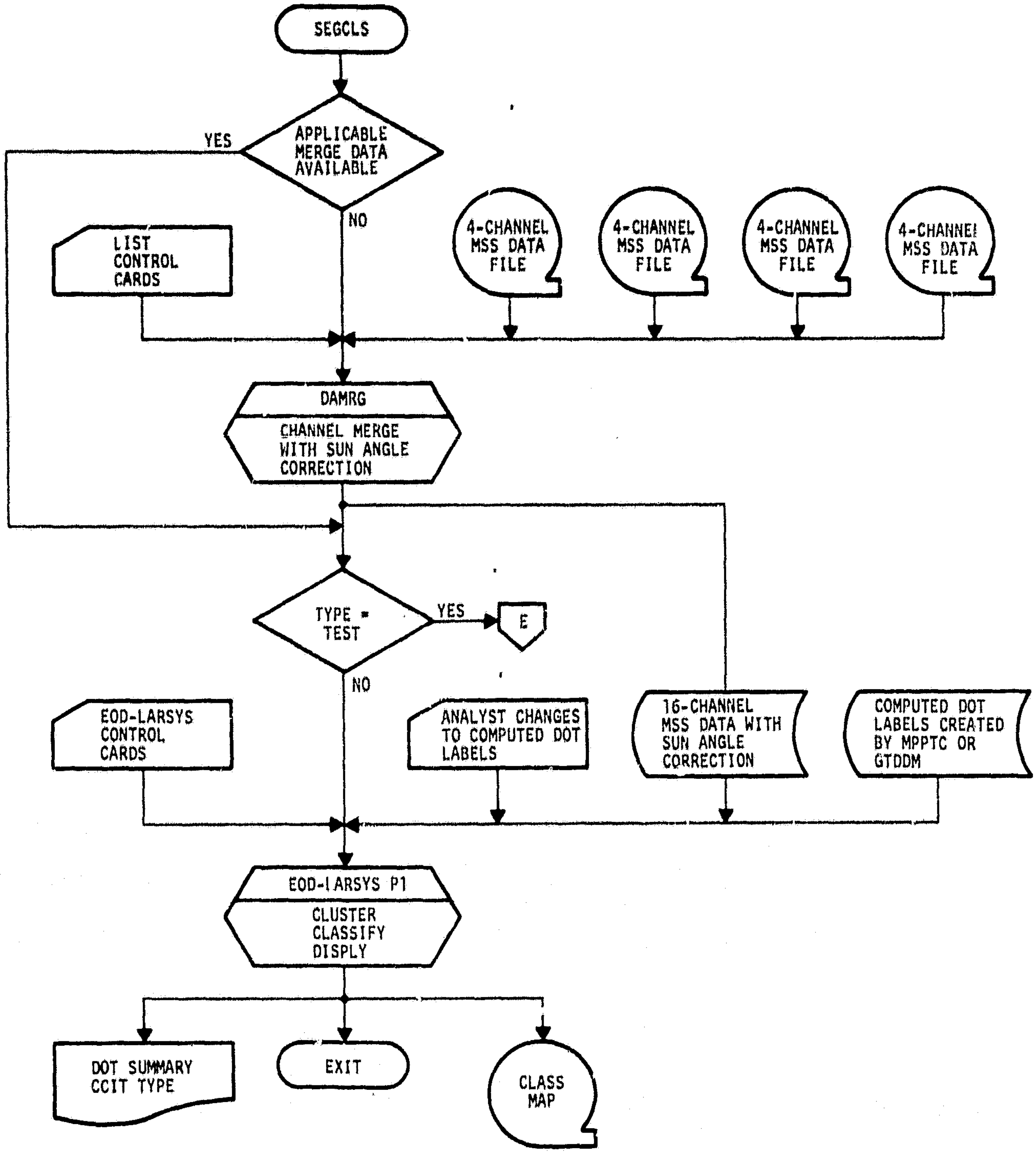
LIST PROCESSOR (Continued)



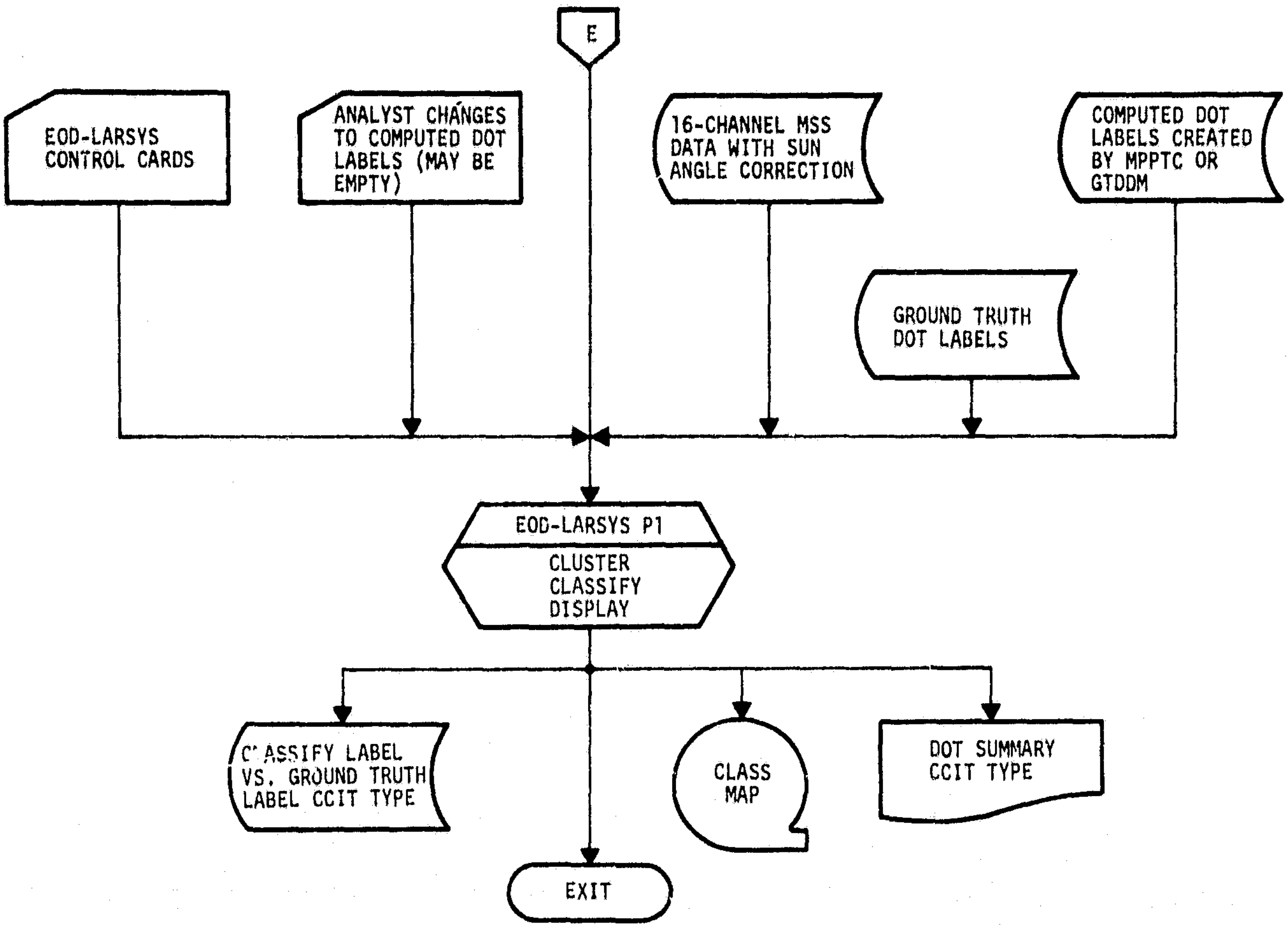
LIST PROCESSOR (Concluded)



SEGMENT CLASSIFICATION



SEGMENT CLASSIFICATION (Concluded)



3.2.2 SOFTWARE COMPONENT NO. 2 (DAMRG)

The DAMRG program furnishes a general capability to merge formatted MSS data file into a file suitable for input to pattern recognition systems such as EOD-LARSYS. Up to six files can be handled in one merge procedure; the option used in LIST processing is four channels from each of four tapes.

This program allows for the following three user options in merging data: channel merge, spatial merge and line merge. The option used to prepare data for LIST processing is the channel merge option whereby pixel data from the specified channels in the user-specified rectangular field are concatenated.

The output file is a Universally or LARSYS III formatted file with a rectangular field with line and sample skip factors set to the value 1 and the first pixel coordinates set to (1,1). Merge channels are renumbered starting with the value 1. Channels are merged in the order of appearance in the input control cards.

Sun angles are extracted from header records or read in from control cards. Gains and biases are unpacked for appropriate channels. Sun angle corrections are performed on option to all pixels making up to the output tape.

A channel merge of Universally formatted output has the sunangles and gains and biases written to the output header; the data and site are also written to the header.

In gains and biases and sunangle extraction, it is assumed that the first channel on any input file is channel number 1.

Reference As-Built Design Specification for a Merging Program for Image Data Files, Aug. 1978, for a detailed description of DAMRG.

3.2.3 SOFTWARE COMPONENT NO. 3 (GTTCN)

The program GTTCN converts Accuracy Assessment Ground Truth Image files in Universal format (351 by 392 in size) to Universal format image files (117 by 196 in size) by mapping six subpixels into one pixel label.

A user supplied control card-image file specifies the format and logical unit number (LUN) and file number of the first file on the input tape, the LUN and the file number of the first file on the output tape, the number of files to be converted, a list option for the crop codes listing and a vector designating which subpixels are to be used in labeling the pixel.

The conversion labels each pixel by majority rule using the user supplied vector to determine which of the subpixels are to be used in the labeling process. Ties are resolved by using the first label.

A Crop Code Listing is producted if requested in the output file.

Reference Preliminary User Guide for the Program GTTCN (Ground Truth Tape Conversion), LEC, July 1978.

3.2.4 SOFTWARE COMPONENT NO. 4 (GTDDM)

The GTDDM program furnishes a general capability to extract a 11 by 19 matrix of dots from a Converted Ground Truth Tape (size 117 by 196) prepared by the GTTCN program. This 11 by 19 matrix is then labelled using the user supplied Crop Code to Category Table in the LARSYS format which is described in the document referenced below.

The 11 by 19 matrix of labelled dots is converted to LACIE format and written to an output file.

A report file is written listing all control cards, the 11 by 19 matrix of labeled dots and the LACIE format dots.

Reference Preliminary User Guide for the Program GTDDM (Ground Truth Dot Dump) for a detailed description of GTTDM.

3.2.5 SOFTWARE COMPONENT NO. 5 (PREPPT)

PREPPT collects data from multiple files, calculates values for each pixel to describe greenness and brightness and writes the LIST variable Input File.

3.2.5.1 Linkages

PREPPT is called by the LISTRR EXEC and calls READID, READTB, READAI, READRD, CALKEY, READGT and WRTQD.

3.2.5.2 Interfaces

Interface is accomplished through common blocks /AI/, /BLØSTG/, /SEASNS/, /SGMENT/.

3.2.5.3 Inputs

List Control Card File (Table 3-1), Biostage Response File (Table 3-2), Seasonal Table File (Table 3-3), AI Response File (Table 3-4), 16-channel MSS Data File, Ground Truth Dot Data File.

3.2.5.4 Outputs

LIST Variable Input File (Table 3-5).

3.2.5.5 Storage Requirement

Not Applicable.

3.2.5.6 Description

PREPPT calls READID to read the types of processing, segment numbers and segment dates. READTB is called to read the seasonal data and biostage data and READAI is called to read the analyst response data and READRD is called to read the 16-channel MSS data file. CALKEY is called to make the calculations described in Appendix D. If testing or training is to be done, READGT is called to read the ground truth data. WRTQD is called to write the LIST Variable Input File

3.2.5.7 Flowchart

N/A

3.2.5.8 Listings

TBD

3.2.6 SOFTWARE COMPONENT NO. 6 (MPPTA)

The program MPPTA writes a loss vector matrix to unit no. 7 to be used by the processor MPPTC.

3.2.6.1 Linkages

The program MPPTA calls subroutines SPPTA, PPTA, CLOCK, GETIME, GETIME, GTDATE, and IDNAME. The subroutine SPPTA uses the function ALPHA and calls the subroutine SETTRM. The Subroutine PPTA calls READIT, NP, ALPHA, and PHI. Subroutines CLOCK, GETIME, GTDATE, and IDNAME are "system subroutines" and descriptive by name.

3.2.6.2 Interfaces

MPPTA interfaces with other routines through calling sequences, and common blocks UN and FV. The common blocks are initialized in PPTBLK.

3.2.6.3 Inputs

The List Variable Disk Files (Table 3-5).

3.2.6.4 Outputs

A loss vector matrix disk file is written.

3.2.6.5 Storage

Program size = 398694.

3.2.6.6 Description

The program MPPTA is the first of two processors used in sequence to classify input data using the Patterson-Pitt-Thandani algorithm for minimum loss classification. MPPTA writes a loss vector matrix to a disk data set to be used by the second processor MPPTC.

3.2.6.7 Flowchart

N/A

3.2.6.8 Listing

N/A

3.2.7 SOFTWARE COMPONENT NO. 7 (MPPTC)

The program MPPTC uses the loss vector matrix file (made by the processor) MPPTA and the pixel data to compute the minimum loss classification.

3.2.7.1 Linkages

The program MPPTC calls subroutines SPPTC, PPTC, CLOSK GETIME, GTDATE, and IDNAME. The subroutine PPTC in turn calls READIT, IBETA, PRBSUM and PHI. The subroutines CLOCK, GETIME, and IDNAME are "system subroutines" and descriptive by name.

3.2.7.2 Interfaces

MPPTC interfaces with other routines through calling sequences, and common blocks UN and FV. The common blocks are initialized in PPTBLK.

3.2.7.3 Inputs

A loss vector disk file.

3.2.7.4 Outputs

A computed dot label file in LACIE format.

3.2.7.5 Storage

Program size = 27550 bytes.

3.2.7.6 Description

The program MPPTC is the second of two processors used in sequence to classify the input data using the Patterson-Pitt-Thandani algorithm for minimum loss classification. MPPTC classifies the data using the loss vector matrix file computed by the first processor.

3.2.8 EOD-LARSYS SYSTEM - DISPLAY PROCESSOR

The EOD-LARSYS system is the main component in the second subsystem of LIST processing. It will accept the merged sun angle corrected MSS data file created by DAMRG together with dot labels computed by the first subsystem routine PPTC. Changes to dot labels corresponding to analyst-identified boundary dots will be input. If the TEST option is invoked, ground truth dot labels prepared by the first subsystem routine GTTDM will also be input. A CCIT - type summary report will be generated as an output product.

In order to achieve the additional - capabilities needed for LIST processing, additions will be made to the DISPLAY processor of the EOD-LARSYS system. These include

1. reading of the ground truth dot labels on the TEST option
2. processing the Type 2 boundary pixels to compute new bias correction variables
3. enhancement of the computation for proportion estimation using Type 2 boundary dots.
4. preparation of the CCIT-type summary report.

3.2.8.1 Linkages

The DISPLAY processor is compiled under FORTRAN IV-G currently implemented on the IBM 370/148 at Purdue-LARS. It is called by routine MONTOR, the driver for the EOD-LARSYS system. It uses several EOD-LARSYS system utility routines, including FSBSFL, WRTFLD, RDDOTS, WRTHED, WRTLN, WRTMTX, FIND12, NUMBER, NXTCHR. It consists of routines DSPLAY, DSPLY1, DSPLY2, REDIF3, SETUP3, EMTHRS, DISTCV, CHIN, CHI, RNORM, TINORM, PCT, DESIG, FLDBOR, PRTSUM, MAPHD, FDSIT, FISHIN, FIST, and PRTPCT. It uses unlabeled common ARRAY, and labeled common blocks /GLOBAL/, /DISPL/.

3.2.8.2 Interfaces

DISPLAY accepts a classification map (MAPTAP) provided by either the CLASSIFY or LABEL processors, a DOTDATA file provided by the DOTDATA processor, and designated fields. It does not provide input to the other processors.

3.2.8.3 Inputs

a. Processor Card

Keyword
$DISPLAY

b. New Control Cards

Keyword	Parameter	Function
GTTR	n	segment number identifying ground truth dot file (needed in test option)
BNRY	(boundary dot name) 1 character	character designating boundary dot name
BIAS	(option number)	specifies method to be used in determining proportion estimates

3.2.8.4 Outputs

A class map (MAPFIL) is written in either Universal or LARSYS III format. Summary reports are output.

3.2.8.5 Storage Requirements

TBD

The summary reports will be augumented with the inclusion of CCIT-type outputs. These outputs include

1. Spatial maps showing dot label vs classification results in the form $name_1/name_2$ for both Type 1 and Type 2 dots. Another set of spatial maps will display dot label vs cluster label of the cluster containing the dot for each dot,

2. bias and boundary dot correction matrices,

3. segment number and acquisition dates of the MSS data from which the computed dot labels were computed.

3.2.8.6 Description

The DISPLAY processor will be augmented to perform the LIST requirements listed above. Most of the changes will involve subroutine DSPLY2.

The following is taken from a Secondary Error Sources report (Techniques Development Section) and will serve to describe the proportion estimation computation using boundary Type 2 dots.

If spatially pure pixels have greater labeling accuracy, then removing the boundary and edge pixels may improve the proportion estimation performance; however, the procedure may become biased from the change in sampling stragegy. This could lead to a deterioration in proportion estimation performance. Thus, real need exists for an empirical test of the two methods, omitting boundary and edge pixels or forcing the analyst to make his best guess. From past experience, a large number of segments will be required to determine the improvement in proportion estimation. As many blind sites as can be assembled can be included in this test.

Actually, these methods of handling boundary pixels are but two methods from a general class of methods. Rather than have an analyst "guess" the label for a boundary pixel or omit the boundary pixel from consideration, another approach is to assume the boundary pixels to be partly small grains. Then the general formula for the small grains proportion estimate follows.

$$\hat{p} = p(1) \left[p(s/1) + \alpha_1 \ p(B/1) \right] + p(2) \left[p(s/2) + \alpha_2 \ p(B/2) \right]$$

α_1, α_2 were zero in DISPLAY

where

$\hat{p}$ = small grains proportion estimate
p(1) = machine estimate of small grains proportion
p(2) = fraction of scene classified other
p(s/i) = probability of pure small grains given class i
p(B/i) = probability of boundary given class i
α_i = fraction of the boundary pixel which is small grains given the boundary pixel is classified into the i^{th} class

For this type of estimator, the analyst would need to label the Type 2 dots as small grains, boundary, or other. The classification is that of Procedure 1 where no boundary dots are used as Type 1 dots (current method).

Suppose there are n bias correction dots (Type 2) of which n_1 are classified into Class 1 and n_2 into Class 2. From the labels the analyst has provided and the classification results, estimates of the parameters in the above formula can be obtained. They are:

$$\widehat{p(s/i)} = \frac{n_{si}}{n_{si} + n_{\phi i} + n_{Bi}}$$

$$\widehat{p(B/i)} = \frac{n_{Bi}}{n_{si} + n_{\phi i} + n_{Bi}}$$

$$p(1) = \frac{N_{n_1}}{\text{Base}} \qquad p(2) = \frac{N_{n_2}}{\text{Base}}$$

where

n_{si} = The number of Type 2 dots which are labeled by the analyst as pure small grains and are classified into the i^{th} class.

n_{Bi} = The number of Type 2 dots which are labeled by the analyst as boundary and are classified into the i^{th} class.

$n_{\phi i}$ = The number of Type 2 dots which are labeled by the analyst as other and classified into the i^{th} class.

N_i = The number of pixels in the scene classified into class i.

Base = 22,932 minus the pixels designated unidentifiable.

The formulation for handling boundary pixels represents an entire class of situations corresponding to different assumptions about α_1 and α_2. The following table shows how the two methods previously discussed along with three more plausible methods lie within this framework.

Assumption	
1. $\alpha_1 = \alpha_2 = 1/2$	Each boundary pixel is considered as being one-half small grains regardless of how it is classified.
2. $\alpha_1 = \alpha_2 = \alpha$ a constant	Each boundary pixel is considered as being the fraction of small grains regardless of how it is classified.
3. $\alpha_1 = \alpha_2$ are constants	The boundary pixels classified into the i^{th} class are considered as being the fraction α_i small grains.
4. $\alpha_i = \dfrac{n_{si}}{n_{si} + n_{pi}}$	This choice for α_i is mathematically identical to omitting boundary pixels. The small grains fraction α_i if the boundary pixels classified into the i^{th} class is considered to be the same as the ratio of pure small grains pixels to all pure pixels for the i^{th} class.
5. α_i is analyst determined for each segment	This is the present method where the analyst is forced to make a decision on each boundary pixel.

For 1, 4, and 5 the value for α_i is apparent. In 2 and 3 the α_i must be estimated. This can be accomplished by using the formula as a regression model where $\hat{p}$ is replaced by the ground truth small grains proportion. This comparison of the assumptions can be performed.

LIST Control Parameters

Beginning Processor	DAMRG, GTTCN, GTDDM, PREPPT, MPPTA, MPPTC or EOD-LARSYS
Ending Processor	DAMRG, GTTCN, GTDDM, PREPPT, MPPTA, MPPTC or EOD-LARSYS

For each segment the following information must be entered:

Type of processing (TRAIN, TEST, CLASSIFY)

Segment number

Segment date

Table 3-1

Robertson Biostage Table

Robertson biostage range	First class response	Second class response
1.0 – 2.0	No vegetation (0)	Green vegetation (1, 2, 3)
2.1 – 2.5	No vegetation or green vegetation (0, 1, 2, 3)	
2.6 – 3.0	Green vegetation (1, 2, 3)	No vegetation (0)
3.1 – 5.0	Green vegetation (1, 2, 3)	
5.1 – 5.5	Green vegetation or turning (1, 2, 3, 4)	Harvested (5)
5.6 – 6.0	Turning (4)	Green vegetation or harvested (1, 2, 3, 5)
6.1 – 6.9	Turning or harvested (4, 5)	
7.0	Harvested (5)	Turning (4)

Table 3-2

FOR WINTER WHEAT SITES (KANSAS)

BIO-LOWER LIMIT	BIO-UPPER LIMIT	MEAN G#	MEAN Br.	STD.DEV. G#	STD.DEV. Br.
0	1.0	1.25	74.38	2.58	10.22
1.1	2.0	(B10-1.0)*12.76/1.1 +(2.1-B10)*1.25/1.1	(B10-1.0)*75.67/1.1 +(2.1-B10)*74.38/1.1	(B10-1.0)*9.55/1.1 +(2.1-B10)*2.58/1.1	(B10-1.0)*17.04/1.1 +(2.1-B10)*10.22/1.1
2.1	2.9	12.76	75.67	9.55	17.04
3.0	3.1	(B10-2.9)*20.16/.3 +(3.2-B10)*12.76/.3	(B10-2.9)*77.06/.3 +(3.2-B10)*75.67/.3	(B10-2.9)*9.37/.3 +(3.2-B10)*9.55/.3	(B10-2.9)*13.25/.3 +(3.2-B10)*17.04/.3
3.2	3.8	20.16	77.06	9.37	13.25
3.9	4.6	(B10-3.8)*21.41/.9 +(4.7-B10)*20.16/.9	(B10-3.8)*67.7/.9 +(4.7-B10)*77.06/.9	(B10-3.8)*7.5/.9 +(4.7-B10)*9.37/.9	(B10-3.8)*8.0/.9 +(4.7-B10)13.25/.9
4.7	5.3	(B10-4.7)*13.66/.7 +(5.4-B10)*21.41/.7	(B10-4.7)*70.3/.7 +(5.4-B10)*67.7/.7	(B10-4.7)*5.7/.7 +(5.4-B10)*7.5/.7	(B10-4.7)*10.8/.7 +(5.4-B10)*8.0/.7
5.4	5.5	13.66	70.3	5.7	10.8
5.6	6.0	(B10-5.5)*5.46/.5 +(6.0-B10)*13.66/.5	(B10-5.5)*89.8/.5 +(6.0-B10)*70.3/.5	(B10-5.5)*7.4/.5 +(6.0-B10)*5.7/.5	(B10-5.5)*27.9/.5 +(6.0-B10)*10.8/.5
6.1	7.0	(B10-6.0)*(-1.36) +(7.0-B10)*5.46	(B10-6.0)*96.24 +7.0-B10)*89.8	(B10-6.0)*3.1 +(7.0-B10)*7.4	(B10-6.0)*12.5 +(7.0-B10)*27.9

Table 3-2

FOR SPRING WHEAT SITES (N.D.)

BIO-LOWER LIMIT	BIO-UPPER LIMIT	MEAN G#	MEAN Br.	ST.DEV. G#	ST.DEV. Br.
0	0.9	1.25	74.38	2.58	10.22
1.0	1.1	(B10-1.0)*12.6 +(1.2-B10)*6.25	(B10-1.0)*290.05 +(1.2-B10)*371.9	(B10-1.0)*15.5 +(1.2-B10)*12.9	(B10-1.0)*74.35 +(1.2-B10)*51.1
1.2	2.2	2.52	58.01	3.1	14.67
2.3	2.4	(B10-2.2)*38.95 +(2.4-B10)*12.6	(B10-2.2)*333.1 +(2.4-B10)*290.05	(B10-2.2)*43.1 +(2.4-B10)*15.5	(B10-2.2)*94.45 +(2.4-B10)*74.35
2.5	2.9	7.79	66.62	8.62	18.89
3.0	3.4	(B10-2.9)+44.83 +(3.5-B10)*12.98	(B10-2.9)*115.72 +(3.5-B10)*111.03	(B10-2.9)*23.27 +(3.5-B10)*14.37	(B10-2.9)*13.6 +(3.5-B10)*31.48
3.5	3.7	26.39	69.43	13.96	8.16
3.8	4.1	(B10-3.7)*65.32 +(4.2-B10)*52.78	(B10-3.7)*139.4 +(4.2-B10)*138.86	(B10-3.7)*23.48 +(4.2-B10)*27.92	(B10-3.7)*22.48 +(4.2-B10)*16.32
4.2	4.6	32.66	69.7	11.74	11.24
4.7	4.9	(B10-4.6)*60.73 +(5.0-B10)*81.65	(B10-4.6)*165.25 +(5.0-B10)*174.25	(B10-4.6)*24.53 +(5.0-B10)*29.35	(B10-4.6)*27.63 +(5.0-B10)*28.1
5.0	5.4	24.29	66.1	9.81	11.05
5.5	6.0	(B10-5.4)*15.28 +(6.0-B10)*40.48	(B10-5.4)*118.02 +(6.0-B10)*110.07	(B10-5.4)*14.98 +(6.0-B10)*16.35	(B10-5.4)*29.65 +(6.0-B10)*18.42
6.1	7.0	(B10-6.0)*4.42 +7.0-B10)*9.17	(B10-6.0)*59.04 +(7.0-B10)*70.81	(B10-6.0)*4.49 +(7.0-B10)*8.99	(B10-6.0)*6.78 +(7.0-B10)*17.79

Table 3-3

AI Response File

Record Type	Record Column	Type of Data
Header 1	2 - 5	Segment Number
	7	Season
	9 - 12 14 - 17 19 - 22 24 - 27	Julian Acquisition Dates
Header 2	2 - 5	Reference Acquisition Date
	11 - 12 15 - 17 20 - 22 25 - 27	Robertson Biostage Numbers for Acquisition Dates in Header 1.
Pixel	(Minimum of 1, maximum of 209 per segment)	
	2 - 3	Line Number
	6 - 7	Pixel Number
	10 - 13	AI Type of Pixel (1 value only)
	15 - 18	Vegetation Indication
	20 - 23	
	25	AI Label

Table 3-4

LIST Variable Input File

File Header Record

    Number of Segments - N
    N Segment Numbers

Segment Header Records

    Segment Header Record 1
    Segment Header Record 2

Pixel Records (Minimum of 1, maximum of 209)

    Line Number
    Column Number
    Ground Truth Category
    AI Label
    Dot Type Label
    AI Vegetation Indication 1 to 4
    AI Comment
    The following elements calculated as described in Appendix D
    where I = Pixel Number
        BRIET(I,J) J=1,4
        GREEN(I,J) J=1,4
        SQAIRB(I)
        SQAIRG(I)
        PIEB(I)
        PIEG(I)
        CANKY(I,J) J=1,4
        CANTJ(I)

Table 3-5

4. OPERATION

The LIST processor, implemented as a system of processors called by an executive file LISTPR, uses the Purdue-LARS 370/148 computer system.

The following input files are required and exact linkages will be furnished in the "As-Built Design Specification for a List Processing System".

Card image file giving type of processing, segment numbers and dates.

4-channel MSS data files (4)

1-channel ground truth file

Robertson Biostage file

Seasonal constents file

AI responses file

MPPTA control cards

MPPTC control cards

Further details to be furnished in the "As-Built Design Specification for a LIST Processing System".

APPENDIX A

A.1 SOFTWARE SUBPROGRAM NO. 1 (READID)

READID reads the LIST control cards.

A.1.1 Linkages

READID is called by PREPPT.

A.1.2 Interfaces

Interface is accomplished through common block /SGMENT/.

A.1.3 Inputs

List Control File giving types of processing, segment numbers and dates.

A.1.4 Outputs

None

A.1.5 Storage Requirement

N/A

A.1.6 Description

READID stores the data from the LIST Control File in common block /SGMENT/.

A.1.7 Flowchart

N/A

A1.1.8 Listings

TBD

A.2 SOFTWARE SUBPROGRAM 2 (READTB)

READTB reads the Seasonal Table File and the Biostage Response File.

A.2.1 Linkages

READTB is called by PREPPT.

A.2.2 Interfaces

Interface is accomplished through common blocks /BIOSTG/, /SEASNS/.

A.2.3 Inputs

Seasonal Table File
Biostage Response File

A.2.4 Outputs

None

A.2.5 Storage Requirement

Not applicable.

A.2.6 Description

The data from the Biostage Response File is stored in /BIOSTG/. The data from the Seasonal Table File is stored in /SEASNS/.

A.2.7 Flowchart

N/A

A.2.8 Listings

TBD

A.3 SOFTWARE SUBPROGRAM NO. 3 (READAI)

READAI reads one set of responses from the AI Response File.

A.3.1 Linkages

READAI is called by PREPPT.

A.3.2 Interfaces

Interface is accomplished through common block /AI/.

A.3.3 Inputs

AI response File.

A.3.4 Outputs

None.

A.3.5 Storage Requirement

Not applicable.

A.3.6 Description

One set of responses from the AI Response File is stored in common block /AI/.

A.3.7 Flowchart

N/A

A.3.8 Listings

TBD.

A.4 SOFTWARE SUBPROGRAM NO. 4 (READRD)

READRD reads one set of data from the 16-Channel MSS Data File.

A.4.1 Linkages

READRD is called by PREPRT.

A.4.2 Interfaces

Interface is accomplished through common block /RAWDTA/.

A.4.3 Inputs

16-Channel MSS Data File.

A.4.4 Outputs

None.

A.4.5 Storage Requirement

Not applicable.

A.4.6 Description

One set of data is read from the 16-Channel Data File.

A.4.7 Flowchart

N/A

A.4.8 Listings

TBD.

A.5 SOFTWARE SUBPROGRAM NO. 5 (CALKEY)

CALKEY calculates greenness and brightness factors.

A.5.1 Linkages

CALKEY is called by PREPPT.

A.5.2 Interfaces

Interface is accomplished through common blocks /BIOSTG/, /AI/, /SEASNS/.

A.5.3 Inputs

See Appendix D.

A.5.4 Outputs

See Appendix D.

A.5.5 Storage Requirement

Not applicable.

A.5.6 Description

See Appendix D.

A.5.7 Flowchart

N/A

A.5.8 Listings

TBD

A.6 SOFTWARE COMPONENT NO. 6 (READGT)

READGT reads one set of ground truth data.

A.6.1 <u>Linkages</u>

READGT is called by PREPPT

A.6.2 <u>Interfaces</u>

Interface is accomplished through common block /GT/.

A.6.3 <u>Inputs</u>

Ground Truth Dot Data File.

A.6.4 <u>Outputs</u>

None.

A.6.5 <u>Storage Requirement</u>

N/A

A.6.6 <u>Description</u>

The ground truth dot data is read into common block /GT/.

A.6.7 <u>Flowchart</u>

N/A

A.6.8 <u>Listings</u>

TBD

A.7 SOFTWARE COMPONENT NO. 7 (WRTQD)

WRTQD writes one record to the LIST Variable Input File.

A.7.1 Linkages

WRTQD is called by PREPPT.

A.7.2 Interfaces

Interface is accomplished through common blocks.

A.7.3 Inputs

A.7.4 Outputs

LIST Variable Input File.

A.7.5 Storage Requirement

Not applicable.

A.7.6 Description

The header record and data files are written to the LIST Variable Input File as described in Table 3-5.

A.7.7 Flowchart

See Appendix A.

A.7.8 Listings

TBD

APPENDIX B

B.1 SOFTWARE SUBPROGRAM NO. 1 (SPPTA)

Subroutine SPPTA reads the input cards and sets option switches for the processor MPTTA.

B.1.1 Linkages

SPPTA is called by the program MPPTA and uses data initialized in PPTBLK. SPPTA uses the function ALPHA and calls the subroutine SETTRM.

B.1.2 Interfaces

SPPTA interfaces with MPPTA through a calling sequence and interfaces with MPPTA and PPTBLK through common blocks UN, PF, TR and FV.

B.1.3 Inputs

Calling sequence: Subr. SPPTA (D, T, ISGZ, NT, E, CATREC, C, N1, N2)

Parameter	Dimension	In/Out	Description
D	1	Out	No. of channel
T	1	Out	No. of classes
ISGZ	26	Out	No. of pixels of acertain
NT	1	Out	Total no. of samples category
E	1	Out	Error Tolerance
C	(10,10)	Out	Cost Matrix
CATREC	26	Out	An array use to indicate which categories are being used.
N1	1	Out	A number that determines certain array sizes

Parameter	Dimension	In/Out	Description
N2	1	Out	A number that determines certain array sizes.

Common Blocks:

See PPTBLK for information about the common blocks.

Input cards (unit NRDR1):

	Variables	Format	Function
1.	PFLAG	15	0- for short printout 1- for printout
2.	D,T,NT	4I5	D- no. channels T- no. of classes (at present T=2) NT- Total no. of samples
3.	ISYM,ICREC	(AI,I3)	Category symbol and associated index
4.	E	F10.7	Error tolerance
5.	((C(I,J)J=1,T),	10F5.2	The cost matrix (one row per card)
6.	IDEF	A1	D- use default data vector input format N- input an input format
7.	(use if IDEF=N) NDATA	I5	Number of data points per pixel
8.	(use if IDEF=n) (IFMT(I), I=1,20)	20A4	Input format
9.	IDEF	A1	D- use default feature index vector N- input a feature index vector

	Variables	Format	Function
10.	(Use if IDEF=N) (FEATVC(I), I-1,D)	30I2	The feature index vector.

B.1.4 Outputs

Input information is printed out.

B.1.5 Storage

Program size = 2694.

B.1.6 Description

SPPTA is the input subroutine for all except the pixel data.

If default options are not used this subroutine inputs the format for the pixel data and the feature index vector.

B.1.7 Flowchart

N/A

B.1.8 Listing

N/A

B.2 SOFTWARE COMPONENT NO. 2 (PPTA)

Subroutine PPTA is the main computational subroutine of the processor MPTTA. Input from SPPTA is passed to PPTA. PPTA with the aid of other subroutines calculates the loss vector matrix and writes it out to unit WUNIT.

B.2.1 Linkages

Subroutine PPTA is called by MPPTA and is passed information from SPPTA. PPTA calls subroutines READIT, PHI, and NP.

B.2.2 Interfaces

PPTA interfaces with other routines through a calling sequence and common blocks UN and PF.

B.2.3 Inputs

Calling sequence:

Subr. PPTA (D, T, ISGZ, NT, N1, N2, PNI, P, Q, R, S, A, E, CATREC, C, X)

Parameter	Dimension	In/Out	Description
D	1	In	No. of channels
T	1	In	No. of classes
ISGZ	26	In	No. of pixels of a certain category
NT	1	In	Total no, of samples
N1	1	In	Dimension for some arrays
N2	1	In	Dimension for some arrays
PNI	N2	In	PN inverse
P	N1	In	Phi function vector
Q	(N1,T)	In	Class phi sum matrix
R	N1	In	PNI*P
S	(N1,T)	In	Working storage

Parameter	Dimension	In/Out	Description
A	(N1,T)	Out	The loss vector matrix
E	1	In	Error Tolerance
CATREC	26	In	Records categories used.
C	(10,10)	In	Cost matrix
X	D	In	The feature vector

Common Blocks:

See PPTBLK for information about the common blocks.

B.2.4 Outputs

The loss vector matrix is printed out and written to unit WUNIT. Optional information is printed out if PFLAG = 1.

B.2.5 Storage

Program size = 6184. bytes

B.2.6 Description

PPTA uses the input of SPPTA and READIT as principle input to compute the loss vector matrix and writes it to unit WUNIT. The method for computing the inverse of matrices in this routine will be changed.

B.2.7 Flowchart

N/A

B.2.8 Listing

N/A

B.3 SOFTWARE PROGRAM NO. 3 (READIT)

Subroutine READIT reads in a vector of data about a pixel, using the input format IFMT, and stores it in the feature vector using the feature index vector.

B.3.1 Linkages

READIT is called by PPTA and PPTC.

B.3.2 Interfaces

READIT interfaces with PPTA and PPTC through a calling sequence and PPTBLK through the common blocks UN, PF, and FV. READIT reads data from unit NRDR2.

B.3.3 Inputs

Calling sequence:

Subr. READIT (LINE, SAMPLE, GTCAT, AICAT, X, ND)

Parameter	Dimension	In/Out	Description
LINE	1	Out	Line number
SAMPLE	1	Out	Sample number
GTCAT	1	Out	Ground Truth
AICAT	1	Out	Category Label
			A. I. Category Label
TYPE	1	Out	Type
X	ND	Out	The feature vector.
ND	1	In	The number of channels.

Common Blocks:

COMMON/FV/FEATVC(30,IFMT(20),NDATA

Parameter	Dimension	In/Out	Description
FEATVC	30	In	The feature index vector
IFMT	2	In	The data input format
NDATA	1	In	Number of data points per pixel

See PPTBLK for information on the other common blocks.

Input cards (unit NRDRZ):

Variables	Format	Function
(XX(I),I=1,ND)	IFMT	Input data for a pixel.

B.3.4 Outputs

If PFLAG = 1 then the vector XX is printed out.

B.3.5 Storage

Program size = 888.

B.3.6 Description

READIT reads in a vector data (length NDATA) about a pixel using the input format IFMT and stores it in the feature vector using the feature index vector as a set of pointers.

B.3.7 Flowchart

N/A

B.3.8 Listing

N/A

B.4 SOFTWARE COMPONENT NO. 4 (PHI)

Subroutine PHI computes the quadratic function vector.

B.4.1 Linkages

PHI is called by subroutines PPTA and PPTC.

B.4.2 Interfaces

PHI interfaces with other routines through a calling sequence.

B.4.3 Inputs

Calling sequence

Subr. PHI(X,P,D,NP)

Parameter	Dimension	In/Out	Description
X	D	In	The feature vector.
P	NP	Out	The phi function vector.
D	1	In	Number of channels.
NP	1	In	Number of terms in the phi vector (N1).

Common blocks:

COMMON/TR/TERMS(300)

Parameter	Dimension	In/Out	Description
TERMS	300	In	Indicates terms of the Phi function to be used.

B.4.4 Outputs

N/A

B.4.5 Storage

Program size = 824. bytes

B.4.6 Description

PHI computes the quadratic function vector. This vector consists of squared terms, cross product terms, first order terms, and one.

B.5 SOFTWARE SUBPROGRAM 5 NO. (NP)

Function NP determines the pointer NP to an upper traiangular array.

B.5.1 Linkages

The function NP is called by the subroutine PPTA.

B.5.2 Interfaces

NP interfaces with PPTA through a calling sequence and as a function subprogram.

B.5.3 Inputs

Calling sequence

Function. NP(I,J,M)

Parameter	Dimension	In/Out	Description
I	1	In	First rectangular coordinate
J	1	In	Second rectangular coordinate
M	1	In	The size of the PN matrix is M by M.

B.5.4 Outputs

N/A

B.5.5 Storage

Program size = 514 bytes

B.5.6 Description

Function NP determines the pointer NP (the function value) to an upper triangular array using the rectangular coordinates I and J.

B.5.7 Flowchart

N/A

B.5.8 Listings

N/A

B.6 SOFTWARE COMPONENT NO. 6 (PPTBLK)

PPTBLK is a block data subprogram. It is used to initialize several variables.

B.6.1 Linkages

N/A

B.6.2 Interfaces

PPTBLK interfaces with the subprograms in this system through the common blocks, FV, UN, LT and TUN.

B.6.3 Inputs

N/A

B.6.4 Outputs

N/A

B.6.5 Storage

Storage = $E4_{16}$ bytes.

B.6.6 Description

PPTBLK is a block data subprogram which initializes the common blocks FV, UN, LT and TUN.

Common blocks:

COMMON/FV/FEATVC(30), IFMT(20),NDATA

Parameter	Dimension	Description
FEATVC	30	The feature index vector.
IFMT	20	The input format for the input data (see READIT)

Parameter	Dimension	Description
NDATA	1	The number of data points per pixel.

The common block UN stores some of the various unit numbers as follows:

NRDL - Card reader for the setup cards or the terminal.

NRDR2 - Card reader for the pixel data.

NPRT - Line printer (or output) unit number.

RUNIT - Utility data set unit number.

WUNIT - Utility data set unit number. (The loss vector is written to this unit)

CØMMØN=LT/LETTER(26)

Parameter	Dimension	Description
LETTER	26	The letters of the alphabet.

The common block TUN stores only the terminal output unit number.

B.6.7 Flowchart

N/A

B.6.8 Listings

N/A

B.7 SOFTWARE SUBPROGRAM NO. 7 (SETTRM)

Subroutine SETTRM records the terms of the Phi function to be used.

B.7.1 Linkages

SETTRM is called by SPPTA and calls the LARSYS functions NXTCHR, FIND12, and NUMBER.

B.7.2 Interfaces

SETTRM interfaces with SPPTA through the callings sequence and common blocks TR and UN.

B.7.3 Inputs

Calling sequence: Subr. SETTRM(D)

Parameter	Dimension	In/Out	Description
D	1	In	No. of channels

Input cards (unit NRDR1):

Keyword	Parameter & default values	Function
ALL	—	All term of the phi function are used.
DEFAULT	—	Squared, linear, and the constant terms are used.
LINEAR	ALL	All linear terms (including the constant) are used.
	or	

Keyword	Parameter & default values	Function
LINEAR	L = N_1, N_2, N_3....	N_i's specify the linear terms used.
SQUARE	ALL	Squared terms are used.
or		
INTERACT	ALL	All interaction terms are used.
or		
INTERACT	I = N_1, N_2	The interaction term between the N_1 term & the N_2 term is used.
CONSTANT	YES	Constant term is used.
or		
CONSTANT	NO	Constant term is not used.
END	—	End of the input information about the Phi function.

B.7.4 Outputs

Common block:

COMMON /TR/TERMS(300)

Parameter	Dimension	In/Out	Description
TERMS	300	Out	Indicates terms of the Phi function to be used.

B.7.5 Storage

N/A

B.7.6 Description

Subroutine SETTRM reads LARSYS type input cards which specify which terms of the phi function are to be used. This information is stored in the array TERMS to be passed back to other subroutines.

B.7.7 Flowchart

N/A

B.7.8 Listing

N/A

B.8 SOFTWARE COMPONENT NO. 8 (NTRM)

Subroutine NTRM computes rectangular coordinates for a location in an upper triangular matrix.

B.8.1 Linkages

Subroutine NTRM is called by SETTRM.

B.3.2 Interfaces

NTRM interfaces with SETTRM through a calling sequence.

B.8.3 Inputs

Calling sequence: Subr. NTRM (D, IJ, I, J)

Parameter	Dimension	In/Out	Description
D	1	In	Dimension of the matrix
IJ	1	In	Upper triangular pointer.
I	1	Out	First rectangular coordinate.
J	1	Out	Second rectangular coordinate.

B.8.4 Outputs

N/A

B.8.5 Storage

N/A

B.8.6 Description

NTRM computes rectangular coordinates from a pointer indicating a location in an upper triangular matrix.

B.8.7 Flowchart

N/A

B.8.8 Listing

N/A

B.9 SOFTWARE COMPONENT NO. 9 (ALPHA)

The function ALPHA returns the number indicating the place in the alphabet that cooresponds to a certain letter.

B.9.1 Linkages

ALPHA is called by subroutines PPTA and SPPTA.

B.9.2 Interfaces

ALPHA interfaces with other routines through a calling sequence.

B.9.3 Inputs

Calling sequence: Function ALPHA(S)

Parameter	Dimension	In/Out	Description
S	1	In	A letter of the alphabet.

B.9.4 Outputs

N/A

B.9.5 Storage

N/A

B.9.6 Description

For an input letter ALPHA returns the number that indicates its order of occurance in the alphabet.

B.9.7 Flowchart

N/A

B.9.8 Listing

N/A

APPENDIX C

C.1 SOFTWARE COMPONENT NO. 1 (SPPTC)

Subroutine SPPTC reads the input cards and sets option switches.

C.1.1 Linkages

SPPTC is called by the program MPPTC and uses data initialized in PPTBLK.

C.1.2 Interfaces

SPPTC interfaces with MPPTA through a calling sequence and interfaces with MPPTC and PPTBLK through blocks UN, PF, and FV.

C.1.3 Inputs

Calling sequence:

Subr. SPPTC (UNIT, M, D, T, ISGZ, NT, CATREC, NP)

Parameter	Dimension	In/Out	Description
UNIT	1	Out	Unit number for the loss vector matrix data set
M	1	Out	First dimension of the loss vector matrix
D	1	Out	Number of channels
T	1	Out	Number of classes
ISGZ	10	Out	Number of classified into the different classes
NT	1	Out	Total number of pixels
CATREC	26	Out	Array recording occurance of categories
N1	1	Out	Array size used in PPTC
NP	1	Out	Same as NT

Common blocks:

See PPTBLK for information about the common blocks.

Input cards (unit NRDR1);

	Variables	Format	Function
1,	PFLAG	I5	0- for short printout 1- for long printout
2.	NT	I5	NT- Total number of pixels
3.	IDEF	A1	D- use default data vector input format N- input an input format
4.	(use if IDEF=N) NDATA	I5	Number of data points per pixel
5.	(use if IDEF=N) (IFMT(I),I=1,20)	20A4	Input format
6.	IDEF	A1	D- use default feature index vector N- input a feature index vector
7.	(use if IDEF=N) (FEATVC(I),I=1,D)	30I2	The feature index vector

C.1.4 Outputs

Input information is printed out.

C.1.5 Storage

Program size = 2694.

C.1.6 Description

SPPTC is the input subroutine for all except the pixel data. If default options are not used this subroutine inputs the format for the pixel data and the feature index vector.

C.1.7 Flowchart

N/A

C.1.8 Listing

N/A

C.2 SOFTWARE COMPONENT NO. 2 (PPTC)

Subroutine PPTC is the main computational subroutine of the processor MPPTC. Input from SPPTC is passed to PPTC. PPTC with the aid of other subroutines calculates the classification losses to find the minimum loss.

C.2.1 Linkages

Subroutine PPTC is called by MPPTC or MPPTCI and is passed information from SPPTC or SPPTCI. PPTC calls subroutines READIT and FHI.

C.2.2 Interfaces

PPTC interfaces with other routines through a calling sequence and common blocks UN and PF.

C.2.3 Inputs

Calling sequence:

Subr. PPTC(M, D, T, ISGZ, NT, UNIT, N1, A, L, P, CATREC, X, NP)

Parameter	Dimension	In/Out	Description
M	1	In	First dimension of the loss vector matrix
D	1	In	Number of channels
T	1	In	Number of classes
ISGZ	10	In	Number of pixels classified into the different classes.
NT	1	In	Total number of pixels
UNIT	1	In	Unit number for the loss vector matrix data set
N1	1	In	Array size for A and P
A	(N1,T)	In	The loss vector matrix.

Parameter	Dimension	In/Out	Description
L	T	-	The losses for each class
P	N1	-	The phi function vector
CATREC	26	In	Array recording occurance categories
X	D	-	The feature vector
NP	1	In	Same as NT

Common blocks:

See PPTBLK for information about the common blocks.

C.2.4 Outputs

Classification information is printed out.

C.2.5 Storage

Program size = 2550.

C.2.6 Description

PPTC takes the interproduct of a loss vector and a phi vector to determine a class loss for a particular feature vector. The minimum of these is used as the classification for a particular set of input data.

C.2.7 Flowchart

N/A

C.2.8 Listing

N/A

C.3 SOFTWARE COMPONENT NO. 3 (IBETA)

The function IBETA returns a letter corresponding to a category index.

C.3.1 Linkages

IBETA is called by subroutines PPTC and PRBSUM.

C.3.2 Interfaces

IBETA interfaces with other routines through a calling sequence and the common block LT.

C.3.3 Inputs

Calling sequence: Function IBETA (CAT, CATREC)

Parameter	Dimension	In/Out	Description
CAT	1	In	Category index.
CATREC	26	In	Array loader with index numbers corresponding to the letters of the alphabet.

Common block:

COMMON /LT/LETTER(26)

Parameter	Dimension	In/Out	Description
LETTER	26	In	An array containing the letters of the alphabet.

C.3.4 Outputs

N/A

C.3.5 Storage

N/A

C.3.6 Description

IBETA returns a letter corresponding to a category index.

C.3.7 Flowchart

N/A

C.3.8 Listing

N/A

C.4 SOFTWARE COMPONENT NO. 4 (PRBSUM)

Subroutine PRBSUM partially compiles and prints tables of joint occurances.

C.4.1 Linkages

PRBSUM is called by PPTC and calls IBETA.

C.4.2 Interfaces

PRBSUM interfaces other routines through calling sequences and the common block UN.

C.4.3 Inputs

Calling sequence: Subr. PRBSUM (PROBT, T, CATREC)

Parameter	Dimension	In/Out	Description
PROB	(11,11)	In	A matrix of joint counts.
T	1	In	The no. of categories.
CATREC	26	In	An array indicating the category index that corresponds to a category.

C.4.4 Outputs

PRBSUM prints out joint occurance tables.

C.4.5 Storage

N/A

C.4.6 Description

PRBSUM fills in the totals and prints out tables of joint occurances.

C.4.7 <u>Flowchart</u>

N/A

C.4.8 <u>Listing</u>

N/A

APPENDIX D

APPENDIX D

The following calculations are made in computing the LIST variables.

BRIET (i,j) = (B(i, j) - SDB)/MEANB

where i = pixel (1-209)

j = index to acquisition number

B(i,j) = value extracted from 16 tape using the KAUTH transformation

SDB is calculated as described in Table 3-3

(BIØ = biostage from second AI header card Table 3-4)

MEANB is calculated as described in Table 3-3

GREEN (i,j) = (G(i,j)-SDG)/MEANB

where i = pixel (1-209)

j = index to acquisition number

G (i,j) = value extracted from 16-channel tape

SDG is calculated as described in Table 3-3

(BIO = biostage from second AI header card Table 3-4)

MEANG is calculated as described in Table 3-3

ABRIET (i,j) = /BRIET(i,j)/

AGREEN(i,j) = /GREEN(i,j)/

$$SQAIRB(i) = \sum_{j=1}^{4} (BRIET(i,j)^2)$$

$$SQAIRG(i) = \sum_{j=1}^{4} (GREEN(i,j)^2)$$

$$PIEB(i) = \prod_{j=1}^{4} (1 + ABRIET(i,j))$$

$$PIEG(i) = \prod_{j=1}^{4} (1 + AGREEN(i,j))$$